MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2 2018/2019

TRT 3241- REAL-TIME SYSTEMS

(All sections / Groups)

14th MARCH 2019
9.00 a.m. – 11.00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This examination paper consists of 6 pages (including the cover page) with 5 questions only.
2. All questions carry equal marks. The allocation of marks for each question is given in the paper.
3. Answer **ALL** questions.
4. Write your answers in the answer booklet provided.

Question 1 [10 marks]

a) Identify the key differences between hard real-time systems, soft real-time systems, and firm real-time systems.

[3 marks]

b) Once a target image gains control from a loading program, software components of the target image will perform initialization on harware, real-time operating system, and application according to the steps depicted in Figure 1 below.

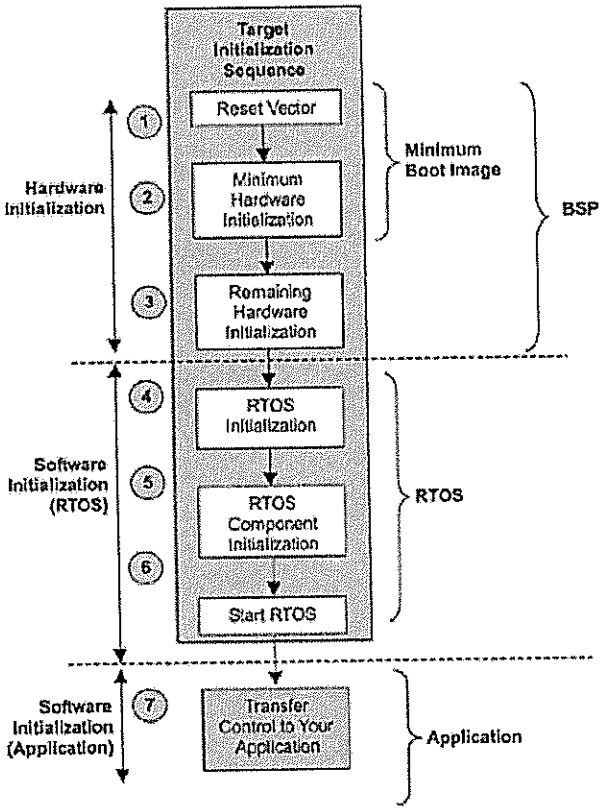

Figure 1: Target image initialization

Explain in detail the steps involved in hardware initialization.

[4 marks]

c) Describe **THREE (3)** components of kernel in real-time operation system (RTOS).

[3 marks]

Continued...

Question 2 [10 marks]

a) Assuming that a task requires a semaphore token to get exclusive access to a shared data structure. While the task is operating on this data structure, the task gets deleted. If not handled appropriately, describe **TWO (2)** consequences of premature deletion of a task.

[2 marks]

b) List **FOUR (4)** typical operations that developers might want to perform with the semaphores in an application.

[2 marks]

c) Given the following condition:

When a message queue is first created, it is in an empty state. If a task attempts to receive a message from this message queue while the queue is empty, the task blocks and, if it chooses to, is held on the message queue's task-waiting list, in either FIFO or priority-based order.

If another task sends a message to the message queue, the message is delivered to the blocked task. The blocked task is then removed from the task-waiting list and moved to either ready or the running state. The message queue in this case remains empty because it has successfully delivered the message. If another message is sent to the same message queue and no tasks are waiting in the message queue's task-waiting list, the message queue's state becomes not empty.

As additional messages arrive at the queue, the queue eventually fills up until it has exhausted its free space. At this point, the number of messages in the queue is equal to the queue's length, and the message queue's state becomes full. While a message queue is in this state, any task sending messages to it will not be successful unless some other task first requests a message from that queue, thus freeing a queue element.

Draw a simple Finite State Machine (FSM) diagram to depict the transitions of states in a message queue.

[3 marks]

d) Signals can be used for synchronization between tasks. They are normally associated with hardware events sent by hardware interrupt service routines (ISRs). State **TWO (2)** reasons why signals should be used with caution.

[2 marks]

e) Explain the main function of TCP/IP protocol stack in RTOS.

[1 mark]

Continued...

Question 3 [10 marks]

a) Define an '*exception*'. Differentiate between synchronous exception and asynchronous exception. Classify the following examples as either synchronous exception or asynchronous exception:

i. An arithmetic operation that results in a division by zero.
ii. Pushing the reset button on the embedded board.

[4 marks]

b) Assume that an application requires one soft timer. The timeout value equals 300ms. The least common denominator is set to 50ms. If each hardware timer tick represents 10ms, calculate the countdown value of the system timer ISR.

The *ISR_timeout_fn* must work with this value and notify the worker task appropriately. Explain how the function performs the task by using the value.

What will be the countdown value for the application in the timer-countdown table based on 100ms granularity?

[4 marks]

c) The individual device drivers provide the actual implementation of each function in the uniform I/O set. The uniform I/O API set can be represented in the C programming language syntax as a structure of function pointers.

Write the C structure defining the uniform I/O API set.

[2 marks]

Continued...

Question 4 [10 marks]

a) One approach to memory management in embedded systems uses the method of fixed-size memory pools. However, the method has some drawbacks. Discuss **TWO (2)** drawbacks.

[2 marks]

b) Table 1 below summarizes the properties of three tasks that are scheduled using rate monotonic priority assignment.

Table 1: Properties of Tasks

Periodic Task	Execution Time	Period (ms)
Task 1	20	100
Task 2	30	150
Task 3	50	300

i. Calculate the processor utilization.
ii. Is the system schedulable?
iii. Will each task meets its deadline?

[3 marks]

c) The critical section of a task is a section of code that access a shared resource. Therefore, it is important to evaluate the criticality of the critical section and the overall system impact before deciding on which mutual exclusion algorithm to use for guarding a critical section.

State **THREE (3)** conditions that must be satisfied by the chosen mutual exclusion algorithm in order to ensure that each task has exclusive access to the shared resource.

[3 marks]

d) In embedded systems, resources are shared among various concurrently executing tasks. There are **TWO (2)** categories of shared resources.

Explain the categories. Provide **ONE (1)** example for each.

[2 marks]

Continued...

Question 5 [10 marks]

The wash-machine waits for the 'start' switch to be pressed. After the user presses the 'start' switch, the machine fills the wash tub with water. The water filling continues until the high level is sensed. The machine starts the agitation motor and continues agitating the wash tub until the user presses the 'stop' switch or the preset timer expires. After the agitation stops, the machine waits for the user to press the 'startDrying' switch. After the user presses the 'startDrying' switch, the machine starts the hot air blower and continues blowing hot air into the drying chamber until the user presses the 'stop' switch or the preset time expires.

Based on the above description, answer the following questions:

a) Categorize the type of real-time system for the wash-machine. Justify your answer.

[1.5 marks]

b) Using an outside-in approach, identify the inputs and outputs of the wash-machine, and draw them in a simple high-level context diagram.

[4.5 marks]

c) The outside-in approach to decomposing an application is an example of one practical way to identify types of concurrent tasks that are dependent on or interact with I/O devices. Further refinement of the diagram would yield additional tasks, hence require more formalized ways to identify concurrency.

Discuss **FOUR (4)** guidelines that you know would be helpful for application designers to identify concurrency in their real-time applications.

[4 marks]

End of Paper.